Exact Approaches for Solving Robust Prize-Collecting Steiner Tree Problems

Eduardo Álvarez-Miranda[a], Ivana Ljubić[b], Paolo Toth[a,1]

[a] *DEIS, Università di Bologna, Bologna, Italy*
[b] *Department of Statistics and Operations Research, University of Vienna, Vienna, Austria*

Abstract

In the *Prize-Collecting Steiner Tree Problem* (PCStT) we are given a set of customers with potential revenues and a set of possible links connecting these customers with fixed installation costs. The goal is to decide which customers to connect into a tree structure so that the sum of the link costs plus the revenues of the customers that are left out is minimized. The problem, as well as some of its variants, is used to model a wide range of applications in telecommunications, gas distribution networks, protein-protein interaction networks, or image segmentation.

In many applications it is unrealistic to assume that the revenues or the installation costs are known in advance. In this paper we consider the well-known Bertsimas and Sim (B&S) robust optimization approach, in which the input parameters are subject to interval uncertainty, and the level of robustness is controlled by introducing a control parameter, which represents the perception of the decision maker regarding the number of uncertain elements that will present an adverse behavior.

We propose branch-and-cut approaches to solve the robust counterparts of the PCStT and the Budget Constraint variant and provide an extensive computational study on a set of benchmark instances that are adapted from the deterministic PCStT inputs. We show how the *Price of Robustness* influences the cost of the solutions and the algorithmic performance.

Finally, we adapt our recent theoretical results regarding algorithms for a general class of B&S robust optimization problems for the robust PCStT and its budget and quota constrained variants

Keywords: Prize Collecting Steiner Trees; Robust Optimization; Interval Uncertainty; Mixed Integer Programming; Branch-and-Cut.

1. Introduction

When defining an expansion plan of a fiber optic network in a given area and for a given planning horizon, a telecommunication company needs to decide to which subset of customers a service should be provided. Thereby, two elements need to be taken into account: potential gains in revenue (that will be referred to as *prizes*) of each customer, and infrastructure *costs* needed to connect them. This problem can be formulated as a network optimization problem called the *Prize-Collecting Steiner Tree Problem* (PCStT). In this paper we will focus on the PCStT and the Budget and Quota constrained variants, under data uncertainty assumption.

When facing strategic decisions modeled by the PCStT, companies should consider the presence of uncertainty in problem parameters as an inevitable feature of the decision-making process. In our particular case, customer revenues and connection costs are uncertain parameters since they are affected by many external economic or even social factors. Consequently, uncertainty in both groups of parameters (or at least one of them) should be part of any decision model in order to obtain reliable and robust solutions from the economic point of view.

[1] Corresponding author: Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna, *Viale Risorgimiento 2, 40136 Bologna, Italia* (Tel: +39-051-2093028, Fax: +39-051-2093073).

In our models, robustness can be seen as a guarantee of protection against data uncertainty. This guarantee is provided by the use of the Bertsimas and Sim (B&S) Robust Optimization (RO) approach, see [10], which entails the adoption of *protection functions* that are included in the objective function and/or constraints. Protection functions depend on both, the uncertainty present in the problem's input parameters and the intuition of the decision maker. These protection functions are all in all what determines the *Price of Robustness*, see [11], which can be defined as the worsening of the economic performance of the solutions while ensuring higher level of robustness in presence of higher levels of uncertainty. The resulting model will be called *robust counterpart* of the original deterministic problem.

The PCStT arises as an important problem in Network Optimization from both the algorithmic and practical points of view (cf. Section 2). Therefore, we believe that studying the robust counterpart of the PCStT will help in solving and better understanding not only the robust PCStT itself, but also other related problems in the area of *robust network optimization*.

In this paper we propose several RO variants of the PCStT and establish some connections between them. As main contribution, we propose three different strategies to exactly solve the Robust PCStT (RPCStT). These exact algorithms are all based on Branch-and-Cut techniques and the differences among them are implied by the underlying mathematical programming formulations and the different cutting-plane techniques. An extensive analysis of computational results is carried out in order to assess the performance of the proposed algorithms and their dependence on the problem parameters, and the nature and characteristics of the obtained solutions. This analysis concerns a qualitative study of the solutions in terms of the *Price of Robustness* and an interpretation and assessment of the different algorithmic performances. To complement this analysis, we also consider a budget-constrained variant of the PCStT and adapt the developed algorithms to solve its robust counterpart.

One of the three proposed branch-and-cut approaches has been computationally tested, only for the robust version of the PCStT, in our preliminary work that appeared in [2].

Structure of the paper. In Section 2, the PCStT is formally defined, a review of the main literature is presented, two important variants of the problem, i.e., Budget and Quota PCStT, are defined, and an integer programming formulation is provided. In Section 3, motivations and alternatives to consider parameter uncertainty are presented with an emphasis on the B&S robust optimization model. Subsequently, different Mixed Integer Programming (MIP) formulations for the robust counterpart of the PCStT and the Budget and Quota Constraint variants are presented. Branch-and-Cut algorithms are presented in Section 4. In Section 5 we present and analyze the computational results obtained for different sets of benchmark instances for the robust counterparts of the PCStT and its budget-constrained variant. In Section 6, our recent theoretical results regarding algorithms for a general class of B&S robust optimization problems, see [3], are adapted for the robust PCStT and its variants. Finally, concluding remarks and paths for future work are presented in Section 7.

2. The Prize Collecting Steiner Tree Problem

The term *Prize Collecting* was used for the first time by Balas, see [7], in the context of the traveling salesman problem. However, it was in [12] where the PCStT has been introduced. It is worth to mention that in [36], Segev studied for first time the closely related Steiner tree problem with node weights. A formal definition of the PCStT can be given as follows.

Given is an undirected graph $G = (V, E)$ with $n = |V|$, $m = |E|$, edge costs $c_e \in \mathbb{R}^{>0}$ for all $e \in E$, and node prizes $p_v \in \mathbb{R}^{\geq 0}$ for all $v \in V$. The PCStT consists of finding a tree $T = (V_T, E_T)$ of G, that minimizes the function

$$f(T) = \sum_{e \in E_T} c_e + \sum_{v \in V \setminus V_T} p_v. \tag{1}$$

For a feasible solution T, function (1) corresponds to the sum of the costs c_e of the edges in the tree, $e \in E_T$, plus the sum of the prizes p_v of the nodes that are *not spanned* by the tree, $v \in V \setminus V_T$;

this definition of the PCStT is known as the Goemans and Williamson PCStT (GW-PCStT) [12]. In the context of the expansion of fiber optic networks mentioned above, graph $G = (V, E)$ is the potential network for which we want to find an expansion plan, so edges $e \in E$ are the possible links with construction costs c_e and nodes $v \in V$ represent customers or street intersections with potential revenues $p_v > 0$ or $p_v = 0$, respectively. By $V_{p_i>0}$ $(n' = |V_{p_i>0}|)$ we will denote the set of *potential customers* and by $V_{p_i=0}$, the set of potential Steiner nodes.

The PCStT can be also defined as the problem of finding a tree T that minimizes

$$f_{NW}(T) = \sum_{e \in E_T} c_e - \sum_{v \in V_T} p_v. \tag{2}$$

Function (2) corresponds to the minimization version of the Net-Worth PCStT (NW-PCStT) which was introduced in [26]. Although functions (1) and (2) are equivalent in the sense that both produce the same optimal solutions, they are not equivalent regarding approximation algorithms, see [26].

Approximation algorithms for the GW-PCStT are discussed in [12], [19], [26] and recently in [4]. Heuristic procedures are implemented in [13], [28] and [35]. The first published work on polyhedral studies for the PCStT is [32], where a cutting plane algorithm is proposed. The cuts are efficiently generated when a violation of a generalized subtour elimination constraint (GSEC) is verified. In [31], a branch-and-cut algorithm based on a directed cut-set MIP formulation has been designed and implemented. Several state-of-the-art methods are combined and pre-processing techniques are used. The proposed procedure has significantly improved the algorithm presented in [32]. The same set of benchmark instances has been solved by two orders of magnitude smaller running times. Optimal solutions have also been achieved for large-scale real-world instances concerning the design of optical fiber networks. Another important algorithmic efforts for the PCStT and some of its variants have been presented in [13], [20], [21] and [22].

In [31] an application of the problem is approached for the first time; the exact algorithm developed in the paper is used to solve real world instances for the design of fiber optic networks of a German city where an existing subnetwork needed to be augmented in order to serve new customers in the most profitable way. Over the last few years various other applications have been studied in which the PCStT has shown to play a crucial role in the modeling process. These problems arise from very different industrial and scientific contexts, showing the potential and versatility of the PCStT as a modeling tool.

Relevant applications of the PCStT are found in Bioinformatics in the context of protein-protein interaction networks (PPIN). In [16], [5], [24] and [6] the PCStT is applied to network optimization problems arising in the analysis of PPIN for different datasets of biological processes. The PCStT is used to model an “inference problem” in order to find, or rather “to infer”, functional modules in PPIN. These networks represent signal pathways (constructed by edges) between proteins or protein complexes (represented by nodes). These biological networks are modeled as a graph $G = (V, E)$, where edge costs c_e represent the confidence of interaction between the source and the target of the given edge e, and node prizes p_v corresponds to the differential expression of node v in the network for a given biological process. In [30], where a survey of models and algorithms for cellular response networks is provided, the PCStT and the algorithm studied in [16] (which is based on the exact approach developed in [31]) are presented as state-of-the-art tools for the detection of response networks in the context of analysis of gene expressions. Recently in [23], the author emphasizes the quality of the results obtained using the PCStT model compared with other modeling and algorithmic approaches for the analysis of signaling networks carried out over different gene databases.

The design of a leakage detection system using the PCStT is performed in [34]. The problem consists of finding the optimal location of detectors in an urban water distribution network so that, given a budget constraint, a desired coverage is provided. The instance considered in the paper corresponds to the urban water distribution network of the city of Lausanne, Switzerland.

In [39], the PCStT is used to efficiently detect region-based objects in the context of image recognition. Nodes v represent superpixels and edges e connect pairs of superpixels that share a boundary. Node prizes p_v represent the contribution of the superpixel to the classifier score, and edge costs are a measure of the probability of two superpixels to belong to the same element. The objective is to find a best-scoring subregion identifying the most likely region of the object of interest. It is important to remark that in [16] and [39] the equivalence between the PCStT and the Maximum-weight connected subgraph problem (MWCS) is exploited to model the particular problem. For more details see [25].

2.1 A Integer Programming Formulation for PCStT

To characterize the set of feasible solutions for the PCStT, i.e., subtrees of G, we consider a directed graph model and use connectivity inequalities to guarantee connectivity of the solution.

We transform the graph $G = (V, E)$ into the directed graph $G_{SA} = (V_{SA}, A_{SA})$. The vertex set $V_{SA} = V \cup \{r\}$ contains the nodes of the input graph G *and an artificial root* vertex r. The arc set A_{SA} is defined as $A_{SA} = \{(r, i) \mid i \in V_{p_i>0}\} \cup A$, where $A = \{(i, j), (j, i) \mid e = \{i, j\} \in E\}$. A subgraph T_{SA} of G_{SA} that forms a directed tree rooted at r such that for each node i in T_{SA} there is a directed path between r and i, is called a *Steiner arborescence* and is a feasible solution of the problem in case there is only one outgoing arc from r. We will use the following notation: A set of vertices $R \subset V_{SA}$ and its complement $\bar{R} = V_{SA} \backslash R$, $R \neq \emptyset$, induce two directed cuts: $\delta^+(R) = \left\{(i, j) \mid i \in R, j \in \bar{R}\right\}$ and $\delta^-(R) = \left\{(i, j) \mid i \in \bar{R}, j \in R\right\}$. Let z_{ij}, $\forall (i, j) \in A$, be a binary variable such that $z_{ij} = 1$ if arc (i, j) belongs to a feasible arborescence T_{SA} and $z_{ij} = 0$ otherwise. Let y_i, $\forall i \in V$, be a binary variable such that $y_i = 1$ if node i belongs to T_{SA} and $y_i = 0$ otherwise. The set of constraints that characterizes the set of feasible solutions of PCStT is given by:

$$\sum_{(j,i)\in\delta^-(i)} z_{ji} = y_i \qquad \forall i \in V_{SA} \backslash \{r\} \tag{3}$$

$$\sum_{(i,j)\in\delta^-(R)} z_{ij} \geq y_k, \quad k \in R, \quad \forall R \subseteq V_{SA} \setminus \{r\}, \quad R \neq \emptyset \tag{4}$$

$$\sum_{(r,i)\in\delta^+(r)} z_{ri} = 1 \tag{5}$$

Let x_e, $\forall e \in E$, be a binary variable such that $x_e = 1$ if edge e belongs to a feasible *subtree* T (induced by T_{SA}) and $x_e = 0$ otherwise. The connection between $\mathbf{x}$ and $\mathbf{z}$ variables is given by

$$x_e = z_{ij} + z_{ji} \quad \forall e = \{i, j\} \in E \tag{6}$$

The corresponding set of feasible solutions satisfying these inequalities is given as:

$$\mathscr{T} = \{(\mathbf{x}, \mathbf{y}) \in \{0, 1\}^{|E|+|V|} \mid \ (\mathbf{x}, \mathbf{y}, \mathbf{z}) \text{ satisfies } (3) - (6) \text{ and } \mathbf{z} \in \{0, 1\}^{|A_{SA}|}\}.$$

Constraints (4), also known as *cut* or *connectivity inequalities*, are the directed counterpart of undirected GSECs used in [32]. They ensure that there is a directed path from the root r to each customer k such that $y_k = 1$. In-degree constraints (3) guarantee that the in-degree of each vertex of the tree is equal to one. The root-out-degree constraint (5) makes sure that the artificial root is connected to exactly one of the terminals. In addition, the following inequalities are used to initialize the MIP model:

$$z_{rj} \leq 1 - y_i, \quad \forall i < j, \quad i, j \in V_{p_i>0} \tag{7}$$

$$\sum_{(j,i)\in\delta^-(i)} z_{ji} \leq \sum_{(i,j)\in\delta^+(i)} z_{ij}, \quad \forall i \in V_{p_i=0}. \tag{8}$$

Constraints (7), the so-called *asymmetry constraints*, ensure that for each feasible solution the customer vertex adjacent to the root is the one with the smallest index. Inequalities (8) are the *flow-balance constraints*, originally introduced for the Steiner tree problem (see [29]). Constraints (7) cut off symmetric solutions, while constraints (8) improve the quality of lower bounds of the Linear Programming (LP) relaxation of the MIP model.

In the remainder, let $T = (V_T, E_T)$ denote the tree induced by a pair $(\mathbf{x}, \mathbf{y})$, such that $E_T = \{e \mid x_e = 1\}$ and $V_T = \{v \mid y_v = 1\}$. For simplicity of notation we state that $T \equiv (\mathbf{x}, \mathbf{y})$.

2.2 Variants of the PCStT: Budget and Quota PCStT

Two well-known variants of the PCStT are the Budget Constrained PCStT (B-PCStT) and the Quota Constrained PCStT (Q-PCStT), which are presented for the first time in [26], where also approximation algorithms and computational studies have been provided.

Given a cost budget B, $B \in \mathbb{R}^{\geq 0}$, representing the maximum total cost allowed for the construction of the solution, the B-PCStT is defined as

$$f_B^*(T) = \min_{T \in \mathscr{T}} \left\{ \sum_{v \in V \setminus V_T} p_v \,\middle|\, \sum_{e \in E_T} c_e \leq B \right\}. \tag{9}$$

Given a prize quota Q, $Q \in \mathbb{R}^{>0}$, representing the maximum total prize allowed to be left out of a solution (or the total prize allowed to be lost), the Q-PCStT is defined as

$$f_Q^*(T) = \min_{T \in \mathscr{T}} \left\{ \sum_{e \in E_T} c_e \,\middle|\, \sum_{v \in V \setminus V_T} p_v \leq Q \right\}. \tag{10}$$

Problem (9) and (10) are natural extensions of the problem that appear in the bi-objective optimization framework. There are two conflicting goals, namely, minimization of the cost and maximization of the profit, and typically, one can solve these problems in iterative frameworks by e.g., the weighted sum approach or ϵ-constrained based approaches (see, e.g., [17]).

3. Formulations for Robust PCStT and its Variants

3.1 Robust Optimization Approaches

In this paper we consider decision-making environments with a lack of complete knowledge about the uncertain state of data and instead of dealing with probabilistic uncertainty (as in *stochastic optimization*, see e.g., [38]) we actually deal with *deterministic uncertainty* [10]. In contrast to probabilistic models, that treat the input parameters as random variables, in the deterministic uncertainty models we assume that the input parameters belong to a known deterministic set. This is in the core of many real world applications and it is the motivation supporting the robust optimization approaches, where the essential objective is to find solutions that will have a *reasonably good* performance (of optimality and/or feasibility) for all possible realizations of the parameter values.

In the last 20 years several RO models have been proposed, corresponding to different motivations and conceptual definitions; for a deep and extensive study on the RO we refer the reader to [9]. In our opinion there are three main characteristics that define the differences among RO models: **(1)** The nature of the input data; whether the data belong to e.g., an ellipsoidal set or polyhedral set, a closed interval, or a set of discrete scenarios; **(2)** If robustness is considered with respect to the value of the objective function (*robust solution*), to the feasibility of the solution (*robust model*) or both; **(3)** The definition of *reasonably good performance* of a solution, which is what determines the main features of the model.

In this paper we consider the RO concept by Bertsimas and Sim (B&S) defined in [10] and [11]. This model is considered as one of the most important references in the field of RO. Regarding the first characteristic mentioned above, this approach tackles interval uncertainty. Regarding robustness, the B&S model allows to find solutions that are robust in terms of optimality and/or feasibility of the solution. The definition of what is a *reasonably good performance* of a solution is given by the protection against a pre-defined number of parameters that might be subject to uncertainty.

In this paper we consider interval uncertainty, which means that associated with each input parameter there is a closed interval with its lower and upper bounds. Formally, in the case of the PCStT, an interval $[c_e^-, c_e^+]$, such that $0 < c_e^- \leq c_e^+$, is associated with each edge $e \in E$, and an interval $[p_v^-, p_v^+]$, such that $0 \leq p_v^- \leq p_v^+$ is associated with each customer $v \in V_{p_i>0}$. To simplify the notation, we will define $0 \leq p_v^- \leq p_v^+$ for all nodes $v \in V$, where $p_v^- = p_v^+ = 0$ for potential Steiner nodes $v \in V_{p_i=0}$. Since we consider deterministic uncertainty, each input parameter can take any value from the corresponding interval without any specific (or known) behavior and independently of the values taken by the other parameters. The lower interval values c_e^- and p_v^- will be referred to as *nominal values*, i.e., they are the values to be considered if the deterministic PCStT is solved. *Deviations* from the nominal values are defined as: $d_e = c_e^+ - c_e^-$, for all $e \in E$ and $d_v = p_v^+ - p_v^-$, for all $v \in V$. In the following we will present two ways to derive mathematical programming formulations for the robust counterpart of the PCStT and of its variants.

The PCStT under interval uncertainty has been considered before in [1]. The authors used an alternative RO model based on a Risk/Cost trade-off concept defined in [14] and provided polynomial time algorithms for solving both the PCStT and its robust counterpart on 2-trees. In this context, our work is complementary since we consider a different RO model and we provide a more general algorithmic framework focusing on graphs with general structure.

3.2 The B&S Robust PCStT

Suppose that a decision maker wants to solve the PCStT in which the input parameters, edge costs and node prizes, are subject to interval uncertainty. In many practical applications it is unlikely that all of edge costs and/or node prizes will present an uncertain behavior at the same time. Therefore, we assume that only a subset of input data is subject to uncertainty, while the remaining parameters are fixed to their *nominal* values. More precisely, the decision maker may assume that only Γ_E edges and Γ_V nodes ($\Gamma_E \in [0, m]$ and $\Gamma_V \in [0, n']$) will be subject to uncertainty, although she/he does not know exactly which they are. Without loss of generality, we will assume that the values of Γ_E and Γ_V are integral.

The essence of the model is to find a solution that is "robust" considering all scenarios in which Γ_E edges and Γ_V nodes present an adverse behavior. If $\Gamma_E = 0$ and $\Gamma_V = 0$, then uncertainty is ignored and the problem to solve is nothing but the nominal problem, whereas if $\Gamma_E = m$ and $\Gamma_V = n'$, i.e., full uncertainty is assumed, the most conservative robust solution is sought.
Considering the general mathematical programming formulation for combinatorial optimization problems with interval uncertainty presented in [10], the B&S RPCStT can be formulated as

$$ROPT(\Gamma_E, \Gamma_V) = \min_{T \in \mathscr{T}} \left\{ \sum_{e \in E_T} c_e^- + \beta_E^*(\Gamma_E) + \sum_{v \in V \setminus V_T} p_v^- + \beta_V^*(\Gamma_V) \right\}, \qquad (11)$$

where

$$\beta_E^*(\Gamma_E) = \max \left\{ \sum_{e \in \tilde{E} \cap E_T} d_e \mid \forall \tilde{E} \subseteq E, |\tilde{E}| \leq \Gamma_E \right\}$$

and

$$\beta_V^*(\Gamma_V) = \max \left\{ \sum_{v \in \tilde{V} \cap \{V \setminus V_T\}} d_v \mid \forall \tilde{V} \subseteq V, |\tilde{V}| \leq \Gamma_V \right\}.$$

These last two functions are the so-called protection functions and they provide robustness to the solutions in terms of protection of optimality in presence of a given level of data uncertainty, represented by Γ_E and Γ_V.

An optimal solution for (11) can be interpreted as the one that minimizes the total nominal cost plus the cost of the maximal Γ_E deviations in the cost of the edges of the solution plus the maximal Γ_V deviations in the prizes of the nodes that are *not* spanned by the solution. If $\Gamma_E = m$ and $\Gamma_V = n'$, the solution will obviously correspond to the optimal (worst-case) deterministic solution in which all edge costs and node prizes will be set to their upper bounds. The flexibility provided by Γ_E and Γ_V is the main advantage of the model from the practical point of view, because it allows the decision maker to include her/his preferences in order to control the level of conservatism of the solutions.

Formulation based on compact robust-constraints:. To find a mixed integer programming formulation for (11), it is necessary to rewrite protection functions $\beta^*_E(\Gamma_E)$ and $\beta^*_V(\Gamma_V)$ using auxiliary variables $u_e \in [0,1]$, $\forall e \in E$ and $u_v \in [0,1]$, $\forall v \in V$, which represent the portion of the corresponding deviation, d_e and d_v respectively, included into the protection function. We thus obtain

$$\beta^*_E(\Gamma_E) = \max\left\{\sum_{e\in E_T} d_e u_e \mid u_e \in [0,1] \;\; \forall e \in E, \sum_{e\in E} u_e \le \Gamma_E\right\} \tag{12}$$

and

$$\beta^*_V(\Gamma_V) = \max\left\{\sum_{v\in V\setminus V_T} d_v u_v \mid u_v \in [0,1] \;\; \forall v \in V, \sum_{v\in V} u_v \le \Gamma_V\right\}. \tag{13}$$

When considering (12) and (13) it is clear that the objective function of (11) contains two non-linear nested maximization problems. To overcome this, one can use strong duality. Let $T^* \equiv (\mathbf{x}^*, \mathbf{y}^*)$ be an optimal tree for (11). Objective functions of problems (12) and (13) can be written as $\sum_{e\in E} d_e x^*_e u_e$ and $\sum_{v\in V} d_v(1 - y^*_v)u_v$, respectively. By strong duality (see, e.g. [10]), we have:

$$\beta^*_E(\Gamma_E) = \min\left\{\theta\Gamma_E + \sum_{e\in E} h_e \mid h_e + \theta \ge d_e x^*_e \text{ and } h_e \ge 0 \; \forall e \in E, \; \theta \ge 0\right\} \tag{14}$$

and

$$\beta^*_V(\Gamma_V) = \min\left\{\lambda\Gamma_V + \sum_{v\in V} k_v \mid k_v + \lambda \ge d_v\left(1 - y^*_v\right) \text{ and } k_v \ge 0 \; \forall v \in V, \; \lambda \ge 0\right\}, \tag{15}$$

respectively. Combining (11), (14) and (15), we can formulate the B&S RPCStT as the following Mixed Integer Programming (MIP) model:

$$ROPT(\Gamma_E, \Gamma_V) = \min \sum_{e\in E} c^-_e x_e + \theta\Gamma_E + \sum_{e\in E} h_e + \sum_{v\in V} p^-_v\left(1 - y_v\right) + \lambda\Gamma_V + \sum_{v\in V} k_v \tag{16}$$

s.t.

$$h_e + \theta \ge d_e x_e, \; \forall e \in E \tag{17}$$

$$k_v + \lambda \ge d_v\left(1 - y_v\right), \; \forall v \in V \tag{18}$$

$$h_e \ge 0 \; \forall e \in E, \; k_v \ge 0 \; \forall v \in V \text{ and } \theta, \lambda \ge 0 \tag{19}$$

$$(\mathbf{x}, \mathbf{y}) \in \mathscr{T}. \tag{20}$$

In this model, variables h_e, k_v, θ and λ are called "robust variables", while constraints (17) and (18) are called "compact robust-constraints" as their number is linear in m and n.

Formulation based on robustness cuts:. One can also use Benders decomposition to project out robust variables from the previous formulation. Since every solution $(\mathbf{x}, \mathbf{y}) \in \mathscr{T}$ is feasible for the robust counterpart of the problem, only Benders *optimality cuts* will be needed to describe the robustness of an optimal solution. These optimality cuts are given by constraints (22) and (23) below:

$$ROPT(\Gamma_E, \Gamma_V) = \min \sum_{e \in E} c_e^- x_e + \Theta + \sum_{v \in V} p_v^- (1 - y_v) + \Lambda \tag{21}$$

s.t.

$$\Theta \geq \sum_{e \in S} d_e x_e, \; \forall S \subseteq E, \; |S| \leq \Gamma_E \tag{22}$$

$$\Lambda \geq \sum_{v \in R} d_v (1 - y_v), \; \forall R \subseteq V, \; |R| \leq \Gamma_V \tag{23}$$

$$\Theta, \Lambda \geq 0 \tag{24}$$

$$(\mathbf{x}, \mathbf{y}) \in \mathscr{T}. \tag{25}$$

In this model, additional variables Θ and Λ and constraints (22) and (23) allow to model the two nested maximization problems $\beta_E^*(\Gamma_E)$ and $\beta_V^*(\Gamma_V)$, respectively. Constraints (22) and (23) are called "robustness cuts". In this model we enforce robustness by working directly on the space of variables $(\mathbf{x}, \mathbf{y})$ at the expense of adding an exponential number of robustness constraints. In Section 4, we will show that these constraints can be separated in polynomial time. In Section 5 we will provide a computational study comparing the practical performance of the compact robust constraints versus these robustness cuts. In [18], the authors have proposed to use robustness cuts for modeling robust linear optimization problems with uncertainty in the constraint parameters.

3.3 The B&S Robust NW-PCStT and equivalences

It is known that for the deterministic case the connection between $f(T)$ and $f_{NW}(T)$ is given as

$$f_{NW}(T) = f(T) - \sum_{v \in V} p_v,$$

i.e., the two formulations of deterministic GW-PCStT and NW-PCStT find the same solution because the sum of node revenues is constant. However, when node revenues are subject to interval uncertainty and a B&S robust solution is sought, this sum is not constant anymore. In this case, the robust counterpart of the NW-PCStT is essentially solving a different problem. To better understand this difference, assume for a moment that edge costs are deterministic. Recall now that in the robust counterpart of the GW-PCStT, nominal values for node revenues are set to conservative lower bounds and, therefore *ROPT* corresponds to a *potential increase of revenues*, which a decision maker can miss. On the other hand, conservative setting for the node revenues in the robust NW-PCStT case is to assume the values are set to their upper bounds, p_v^+, for all $v \in V$.

By following the same ideas presented above for the GW-PCStT, the B&S Robust counterpart of the NW-PCStT is defined as:

$$ROPT_{NW}(\Gamma_E, \Gamma_V) = \min_{(\mathbf{x},\mathbf{y}) \in \mathscr{T}} \left\{ \sum_{e \in E} c_e^- x_e + \beta_E^*(\Gamma_E) - \left(\sum_{v \in V} p_v^+ y_v - \eta_V^*(\Gamma_V) \right) \right\} \tag{26}$$

where

$$\eta_V^*(\Gamma_V) = \max \left\{ \sum_{v \in V} d_v u_v \mid \sum_{v \in V} u_v \leq \Gamma_V, \; u_v \in [0,1] \; \forall v \in V \right\}.$$

In other words, when assuming deterministic edge costs, $ROPT_{NW}$ corresponds to a *potential decrease of revenues*, that the decision maker can experience. It can be easily seen from (26) that larger values of Γ_V will increase the total value of the solution (i.e., decrease the total revenue) as it is expected in this RO model. A MIP formulation can be obtained accordingly by following the same procedure explained for the GW-PCStT.

Despite the fact that these two robust formulations essentially model different problems, the next result shows that in particular cases the two formulations are the same.

Observation 1. *For a fixed value of $\tilde{\Gamma}_E \in [0, m]$, and $\Gamma_V \in \{0, n'\}$, the robust counterparts of the GW-PCStT and of the NW-PCStT are equivalent, i.e., they produce identical optimal subtrees. The following connection exists between the corresponding objective values:*

$$ROPT_{NW}(\tilde{\Gamma}_E, 0) = ROPT(\tilde{\Gamma}_E, n') - \sum_{v \in V} p_v^+$$

and

$$ROPT_{NW}(\tilde{\Gamma}_E, n') = ROPT(\tilde{\Gamma}_E, 0) - \sum_{v \in V} p_v^- .$$

3.4 The B&S Robust B-PCStT and Q-PCStT

In the case of both the GW-PCStT and the NW-PCStT, uncertainty is present only in the coefficients of the objective function, which means that their robust counterparts provide protection with respect to the optimality of the solutions. However, in the case of the B-PCStT and if the Q-PCStT, the presence of uncertainty in edge costs and in node prizes affects not only their corresponding objective functions but also their budget and quota constraints, respectively. Therefore, for a given level of uncertainty, the robust counterpart of these problems should not only provide protection in terms of optimality but also in terms of feasibility.

Adopting the ideas presented in the previous sections, the Robust B&S Budget Constrained PCStT (B-PCStT), is defined as:

$$ROPT_B = \min_{(\mathbf{x},\mathbf{y}) \in \mathscr{T}} \left\{ \sum_{v \in V} p_v^- (1 - y_v) + \beta_V^*(\Gamma_V) \,\middle|\, \sum_{e \in E} c_e^- x_e + \beta_E^*(\Gamma_E) \leq B \right\}.$$

According to the previous section, for a given description $\mathscr{T}$ of the deterministic problem, one can consider four possible ways to derive a valid MIP model for this robust counterpart of the problem. The objective function can be modeled using compact or Benders robust constraints. But also the budget constraint can be modeled using one or the other variant. To model the budget constraint using Benders reformulation, we will need to insert the following family of inequalities into the MIP:

$$\sum_{e \in E} c_e^- x_e + \sum_{e \in S} d_e x_e \leq B \quad \forall S \subseteq E, \ \ |S| \leq \Gamma_E \tag{27}$$

These cuts are similar to (22) (see also [18]).

Similarly, the Robust B&S Quota Constrained PCStT (Q-PCStT), is defined as:

$$ROPT_Q = \min_{(\mathbf{x},\mathbf{y}) \in \mathscr{T}} \left\{ \sum_{e \in E} c_e^- x_e + \beta_E^*(\Gamma_E) \,\middle|\, \sum_{v \in V} p_v^- (1 - y_v) + \beta_V^*(\Gamma_V) \leq Q \right\}$$

and again one can consider four ways of deriving a MIP model for this problem.

4. Branch-and-Cut Algorithms

The MIP formulations considered throughout this paper cannot be solved directly, even for small instances, since there is an exponential number of connectivity constraints of type (4) and, in addition, if Benders cuts are used to model the protection functions, there is also an exponential number of robustness cuts to be considered. Consequently, more sophisticated and specific techniques should be designed and implemented to solve these models.

In this section we propose three ways to develop a branch-and-cut (B&C) algorithm for solving the robust PCStT and its budget and quota constrained variants. We will explain the main ideas for solving the RPCStT, and a similar scheme needs to be applied in order to solve the B-RPCStT or the Q-RPCStT.

B&C with Compact Robust Constraints (Compact):. In this approach, we are solving the MIP model in which the deterministic model (3)-(8) is extended by a compact set of auxiliary variables and constraints (17)-(19) that model the protection functions (cf. Section 3.2). In this approach, only connectivity constraints will be separated within a B&C framework. The separation algorithm is an adaptation of the exact approach presented in [31]. The MIP initially contains all variables and the constraints (3), (5)-(8). In addition, we explicitly insert the subtour elimination constraints of size 2:

$$x_{ij} + x_{ji} \leq y_i, \quad \forall i \in V_{SA} \setminus \{r\}, \quad j \in \delta^+(i)$$

to avoid too frequent calls of the maximum flow procedure. The connectivity constraints are separated within the B&C framework by means of the maximum flow algorithm given in [15]. This separation randomly selects a terminal $i \in V_{p_i>0}$, calculates the maximum flow between an artificial root and i and inserts the corresponding (4), if violated. Instead of adding a single violated cut per iteration, we use *nested*, *back-flow* and *minimum cardinality* cuts to add as many violated cuts as possible (see [29] for details). We restrict the number of inserted cuts within each separation callback to 25.

B&C with Separation of Robustness Cuts (R-Cuts):. In this approach, we consider the MIP model in which protection functions are modeled by means of robustness cuts of type (22) and (23). We initialize the model using only the following bounds for Θ and Λ variables:

$$\Theta \leq \sum_{e \in S^*_{\Gamma_E}} d_e \quad \text{and} \quad \Lambda \leq \sum_{e \in S^*_{\Gamma_V}} d_v$$

where $S^*_{\Gamma_E}$ ($S^*_{\Gamma_V}$) is the subset of edges (nodes) containing Γ_E (Γ_V) edges (nodes) with largest deviations. The correctness of the bounds comes from the fact that both Θ and Λ accumulate the deviations of the nominal costs for the solution edges and for the nodes left out of the solution, respectively.

The separation problem for robustness cuts of type (22) is as follows: given the current LP solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\Theta}, \hat{\Lambda})$, find a set $S \subseteq E$ such that $|S| \leq \Gamma_E$ and $\sum_{e \in S} d_e \hat{x}_e$ is a maximum. Assume that a subset of edges S^* satisfies these properties. If $\sum_{e \in S^*} d_e \hat{x}_e > \hat{\Theta}$, the current LP solution violates constraint (22) and hence we insert the cut $\Theta \geq \sum_{e \in S^*} d_e x_e$ into the model. To determine the set S^*, we associate with each edge $e \in E$ a *weight* $w_e = d_e \hat{x}_e$. The separation problem consists in finding the subset of edges of size Γ_E with the maximum weight, which can be done in $O(|E|)$ time (see also [18]). This idea was first implemented in [18] in the context of robust optimization for linear and integer programming under uncertainty. The authors report a remarkable improvement in the running times when using these robustness cuts in the formulations and separation framework instead of a compact formulation.

Robustness cuts are added on the fly, within the B&C framework, i.e., we are not waiting to find an LP-solution that satisfies all the connectivity cuts. Instead, within one separation callback, we insert all the violated connectivity cuts detected plus the (one or two) robustness cuts associated with (22) and (23).

B&C with Separation of Robust Compact Constraints (C-Cuts):. We have observed that not all of compact constraints associated with a protection function are tight in an optimal solution. On the other hand, when the number of nodes and/or edges increases, the size of the compact block of constraints associated with $\beta^*_V(\Gamma_V)$ or $\beta^*_E(\Gamma_E)$ may become a bottleneck of the implementation. Therefore, instead of inserting all these constraints at once, we propose to separate them within a B&C framework. We start with an LP model in which there are no constraints associated with robust variables, except the following ones:

$$\sum_{e\in E} h_e + \theta \leq \sum_{e\in S^*_{\Gamma_E}} d_e \quad \text{and} \quad \sum_{v\in V} k_v + \lambda \leq \sum_{v\in S^*_{\Gamma_V}} d_v$$

and

$$\theta \leq d^*_e \text{ and } \lambda \leq d^*_v$$

where $S^*_{\Gamma_E}$ ($S^*_{\Gamma_V}$) has been described above, and d^*_e and d^*_v are the largest edge and node deviations, respectively.
The separation of constraints (17) can be stated as follows: given an LP-solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{h}}, \hat{\mathbf{k}}, \hat{\theta}, \hat{\lambda})$, find a set $\hat{E} \subseteq E$ of maximum cardinality for which $\hat{h}_e + \hat{\theta} < d_e \hat{x}_e \ \forall e \in \hat{E}$ and insert the corresponding constraints of type (17). Of course, the separation of constraints (17) and (18) can be performed in $O(|E|)$ and $O(|V_{p_i>0}|)$ time, respectively.

Within the B&C framework we first separate all the connectivity constraints (4), and once we find an optimal LP solution, we find a subset of violated compact robust constraints, and insert all of them at once into the current LP.

5. Computational Results

Benchmark Instances. In our computational experiments four sets of benchmark instances have been tested: C, D, K and P. These instances have been used in most of the papers discussing algorithm design for the PCStT ([32],[31],[35]). Instances of group P were introduced in [26] – they are unstructured and designed to have constant node degree and a constant prize/cost ratio. Group K are randomly generated geometric graphs designed to have a structure similar to street maps [26]. Groups C and D were presented in [13]. These two groups of instances are derived from the instances of the Steiner tree problem provided in the OR-Library [8].

Groups C and D are composed by 40 instances with 500 and 1000 nodes respectively, and the number of edges goes from 625 to 12500 and from 1250 to 25000 respectively. Group P is composed by 11 instances with 100, 200 and 400 nodes and the number of edges goes from 300 to 1185. Finally, group K is formed by 23 instances with 100, 200 and 400 nodes and the number of edges goes from 344 to 1493. For more details on the description of instances see Table 7 in the on-line supplement.

Given an original instance Prob for the deterministic PCStT, the corresponding robust instance, named Prob-α-β, ($\alpha \in [0,1]$ and $\beta \in [0,1]$) is derived as follows: the number of nodes and edges are left unchanged. Lower limits for intervals defining edge costs and node prizes are set to the corresponding deterministic values c_e and p_v, i.e., $c^-_e = c_e \ \forall e \in E$ and $p^-_v = p_v \ \forall v \in V$. The upper limit of edge costs, c^+_e is set to $(1+\alpha)c_e \ \forall e \in E$. Similarly, the upper limit of node prizes, p^+_v, is set to $(1+\beta)p_v \ \forall v \in V$. Parameters α and β allow to control the width of the corresponding intervals and, consequently, the level of uncertainty of the problems.For our experiments we consider $\alpha = \beta = 0.05$, which means that both edge costs and node prizes present a deviation equal to the 5% of their corresponding nominal values. In preliminary experiments we also considered deviations of 1% and 2.5%, however, these instances did not allow to clearly show the impact of considering higher levels of uncertainty on both the solution structure and the algorithm performance. A deviation of 5% is in the middle of the values considered in most of the literature which range from 1% up to 10% (see [10], [11] and [18], among other papers). There are other criteria to produce interval data instances from deterministic instances; for example, see [33].

Reduction Tests. Reduction tests for the deterministic PCStT have been implemented in [13], [32], [31] and [37]. It has been demonstrated that the utilization of some of these preprocessing procedures can lead to remarkable improvements of the algorithmic performance. For our interval data instances we have adapted one of these reduction tests, which is described in the following.

Robust Least-cost Test. Let $\mathrm{SP}_{ij}(\Gamma_E)$ be the cost of the B&S robust shortest path between a pair of nodes i and j calculated for Γ_E in G. If there is an edge e connecting i and j such that $\mathrm{SP}_{ij}(\Gamma_E) \leq c_e^-$, then edge e can be eliminated from G.

Since the calculation of $\mathrm{SP}_{ij}(\Gamma_E)$ requires $O(m)$ shortest path calculations (cf. Bertsimas and Sim [10]), in our implementation we have used only a weaker variant of this test in which $\mathrm{SP}_{ij}(\Gamma_E)$ is replaced by $\mathrm{SP}_{ij}(|E|)$. Although somehow conservative, this reduction criterion provides a unique reduced graph valid for any value of $\Gamma_E < |E|$ when solving the RPCStT or any of its variants. For larger instances, the reduced graphs have less than 50% of the original number of edges. It is important to observe that applying this test requires only a few seconds even for large instances. It turned out that the other robust reduction tests cannot be easily derived from their deterministic counterparts – an illustrative example is a degree two test on a potential Steiner node. After merging two edges and two intervals into one, we basically obtain a new edge whose interval contains an extra break point that is needed to model 0, 1 or 2 deviations from the nominal edge costs.

Machine and Implementation. All the experiments were performed on a Intel Core2 Quad 2.33 GHz machine with 3.25 GB RAM, where each run was performed on a single processor. The Branch-and-cut algorithms are implemented using CPLEX 12.2 and Concert Technology. All CPLEX parameters were set to their default values, except the following ones: (i) Branching: we set the highest branching priorities to variables y_v, $v \in V_{p_i>0}$; (ii) Emphasis: this parameter was set to *optimality.* (iii) Maximum Running Time was set to 500 seconds.

In the following tables and figures, the running times are expressed in CPU seconds.

5.1 The Price of Robustness

As mentioned above, the *Price of Robustness* corresponds to the increase of the cost of a robust solution with respect to the nominal cost when increasing the level of robustness, i.e., when increasing the values of Γ_E and Γ_V. In Table 1 we report the relative increase of the cost of the solutions, $\Delta ROPT(\%)$, for different combinations of Γ_E and Γ_V. $\Delta ROPT(\%)$ is defined as $(ROPT - OPT) * 100/OPT$, where $ROPT$ and OPT are the optimal values[2] of the robust and of the nominal solution, respectively. We consider 16 settings obtained by combining $\Gamma_V, \Gamma_E \in \{0, 5, 20, 50\}$. Since we chose $\alpha = \beta = 0.05$ for generating the instances, we would expect $\Delta ROPT(\%)$ to be always not greater than 5%. The difference between 5% and $\Delta ROPT(\%)$ can be seen as the *level of protection* provided by the robust model and the chosen values of Γ_E and Γ_V. From the information reported in Table 1, two main observations can be made: (i) the B&S model seems to provide more protection against uncertainty to groups C and D than to groups K and P, and (ii) in the case of groups C, D and P, parameter Γ_E has a stronger impact on the price of robustness than Γ_V, while in the case of group K, parameter Γ_V is the one with a stronger influence on the price of robustness.

Both observations can be explained by considering the relation between the particular values of Γ_E and Γ_V and the size of the obtained solutions, whose statistics are given in Table 7 in the on-line supplement. In the case of C, D and P instances, the mean number of edges in the solutions is almost always greater than the chosen values of Γ_E, which means that in many cases the cost of some edges in the solution will remain within the corresponding lower limit, which explains why, for a given Γ_V, the average value of $\Delta ROPT(\%)$ does not reach 5% even when $\Gamma_E = 50$. When comparing the mean number of customer nodes in the instances and the mean number of customer nodes connected by the solutions for groups C and D (see Table 7), it can be easily seen that, on average, many customer nodes are connected, which means that the number of non-connected customers, i.e., those nodes whose prizes and deviations are added in the objective funcion, is generally smaller than $\Gamma_V = 50$. This explains why, for a given value of Γ_E, a variation of Γ_V does not strongly increase the value of $\Delta ROPT(\%)$.

[2] In case that none of the exact approaches was able to find an optimal solution within the specified time limit, we used the final upper bound to calculate $\Delta ROPT(\%)$, which is a good approximation considering the quality of the gaps.

		$\Delta ROPT(\%)$											
		C			D			K			P		
Γ_E	Γ_V	min	mean	max	min	mean	max	min	mean	max	min	mean	max
0	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	5	0.00	0.71	5.00	0.00	0.62	5.00	0.67	2.02	3.74	0.36	0.66	1.74
0	20	0.00	0.95	5.00	0.00	0.81	5.00	1.79	3.54	5.00	0.45	0.94	1.89
0	50	0.00	1.11	5.00	0.00	0.93	5.00	1.79	4.07	5.00	0.45	0.98	1.89
5	0	0.00	0.78	3.00	0.00	0.60	3.10	0.00	0.39	1.73	0.46	0.96	1.92
5	5	0.15	1.49	5.00	0.03	1.22	5.00	0.98	2.42	4.41	0.89	1.65	2.87
5	20	0.15	1.72	5.00	0.05	1.41	5.00	2.50	3.99	5.00	1.25	1.93	3.03
5	50	0.15	1.89	5.00	0.06	1.54	5.00	2.67	4.52	5.00	1.25	1.96	3.03
20	0	0.00	1.82	5.00	0.00	1.46	5.00	0.00	0.65	1.98	1.38	2.45	4.00
20	5	0.43	2.53	5.00	0.19	2.16	5.00	0.98	2.72	4.84	1.83	3.18	4.74
20	20	0.43	2.77	5.00	0.20	2.34	5.00	3.05	4.30	5.00	2.23	3.47	5.00
20	50	0.43	2.93	5.00	0.19	2.48	5.00	3.78	4.88	5.00	2.23	3.50	5.00
50	0	0.00	2.39	5.00	0.00	1.88	5.00	0.00	0.73	2.51	2.50	3.34	4.44
50	5	0.72	3.12	5.00	0.47	2.62	5.00	0.98	2.81	4.84	3.01	4.10	5.00
50	20	0.97	3.36	5.00	0.47	2.81	5.00	3.05	4.40	5.00	3.49	4.40	5.00
50	50	0.98	3.53	5.00	0.47	2.95	5.00	4.92	4.99	5.00	3.49	4.44	5.00

Table 1: Basic statistics of $\Delta ROPT(\%)$ (*Price of Robustness*) for different values of Γ_E and Γ_V, groups C, D, K and P.

In the case of group P, particularly for instances P400.{0-4}, the ratio between the connected versus non-connected customers might be a little bit smaller that in the case of C and D, which explains why $\Delta ROPT(\%)$ can be as high as 4.44% for the maximum values of Γ_E and Γ_V.

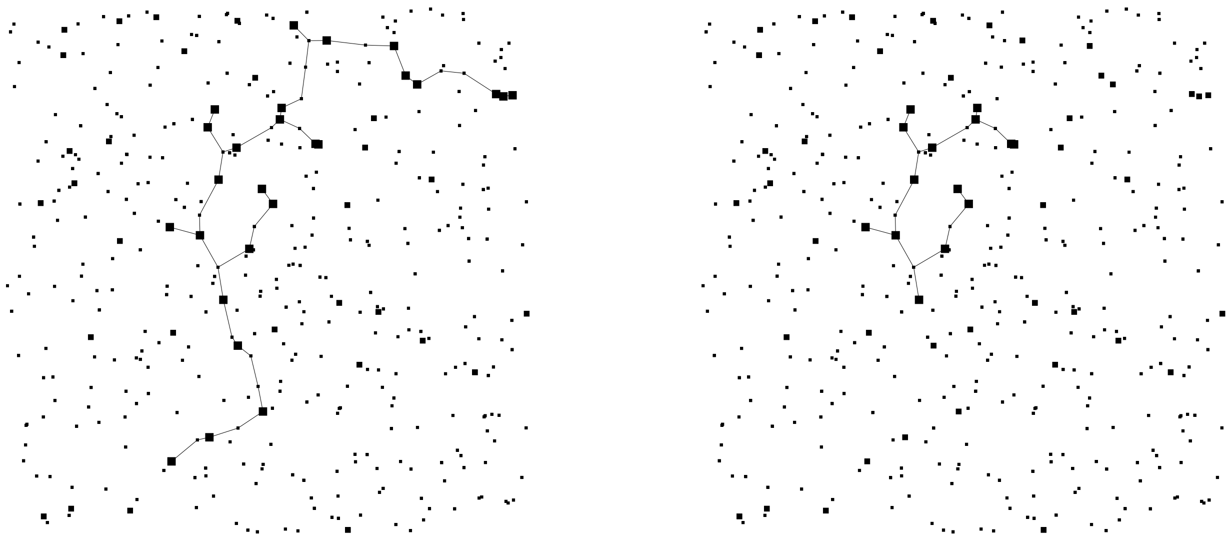
In contrast to what happens for C, D and P groups, in the case of instances of group K, most of the solutions are on average relatively small, which explains why the mean value of $\Delta ROPT(\%)$ can reach almost 5% for large values of Γ_E and Γ_V. The particular Euclidean geometric topology of these instances might also give hints to understand these results; nodes are "locally connected" within a neighborhood, so despite the increase in the prize of non-connected nodes these are not reached because there are no direct connections between a given component and these attractive nodes, which increases the overall cost of the solution.

To look deeper into the impact of Γ_E and Γ_V on the structure of the solutions, Figures 1(a) and 1(b) show two optimal solutions obtained for the instance K400.4-0.05-0.05 for $\Gamma_E = 0$, $\Gamma_V = 50$ and for $\Gamma_E = 50$, $\Gamma_V = 0$, respectively. The $ROPT$ value of the first solution is 403 036 while the cost of the second is 393 919, which represents a relative difference of only 2.3% although the structure of the solutions are quite different; just as a reference, the value of $ROPT$ for $\Gamma_E = 0$ and $\Gamma_V = 0$ is 389 451. These two figures put in evidence the capability of the B&S model to produce very different robust solutions for different levels of conservatism, and, at the same time, to provide a guarantee of protection in terms of the relative increase of the solution cost. This important feature of the model offers the possibility to choose a solution according the perception of the uncertain state of the decision-making environment.

5.2 Algorithmic performance

As mentioned before, to solve the RPCStT we used three different B&C strategies: Compact, Robustness-Cuts and Compact-Cuts. The performance of these different approaches depends not only on the instance group, and the size of the instances therein, but also on the particular selection of the parameters Γ_E and Γ_V.

Figure 2(a) shows the cumulative percentage of instances of group C solved to optimality within a fixed time limit of 500 seconds. We compare the three different approaches for 16 settings across all



(a) $\Gamma_E = 0$ and $\Gamma_V = 50$

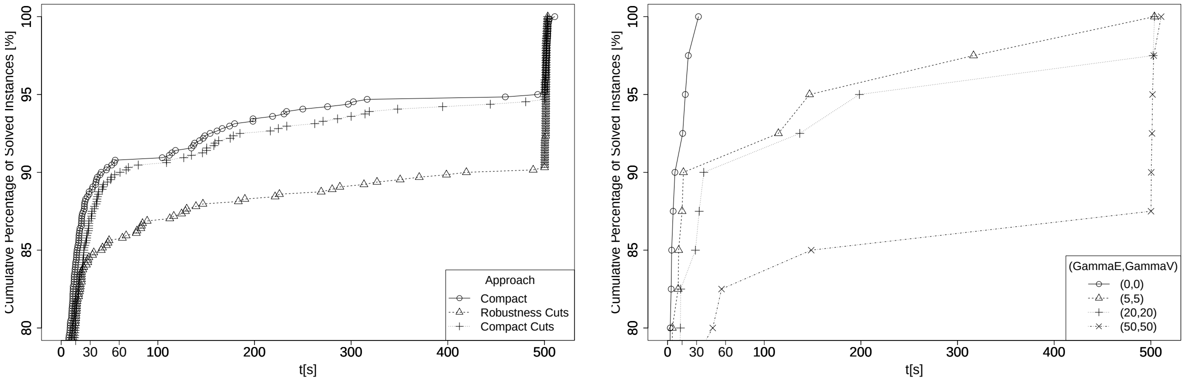
(b) $\Gamma_E = 50$ and $\Gamma_V = 0$

Figure 1: Optimal solutions for the instance `K400.4-0.05-0.05`

values of $\Gamma_E, \Gamma_V \in \{0, 5, 20, 50\}$ and across all the 40 instances of group `C`. From this figure we conclude that the Compact strategy seems to be the best approach for this group since a larger percentage of instances can be solved within smaller running times than those of the other two approaches. However, we also observe that Compact-Cuts approach behaves similarly. To solve 90% of the instances, the Compact approach requires less than 30 seconds, Compact-Cuts slightly more than 30 seconds, and Robustness-Cuts more than 400 seconds. To solve an extra 5% of instances, the Compact approach requires about 300 seconds, while both Compact-Cuts and Robustness-Cuts reach the time limit (500 seconds) for most of them. Overall, the Robustness-Cut approach presents a performance clearly worse than that of the Compact-Cuts, and the Compact-Cuts is outperformed by the Compact approach.

To complement the previous analysis, Figure 2(b) shows the cumulative percentage of solved instances considering four different combinations of Γ_E and Γ_V for the Compact approach only. We can see that for the nominal case ($\Gamma_E = 0$ and $\Gamma_V = 0$), the Compact approach can solve to optimality all the instances within just a few seconds. However, when increasing the values of Γ_E and Γ_V, the running times begin to increase quickly, and even for $\Gamma_E = 5$ and $\Gamma_V = 5$ there are a few instances that cannot be solved to optimality within 500 seconds. Further increasing of the values of Γ_E and Γ_V produces a severe deterioration of the algorithmic performance. For example, when taking $\Gamma_E = 50$ and $\Gamma_V = 50$, almost 15% of the instances can not be solved to optimality within the given time limit. Hence, this is another aspect of the *price of robustness*: obtaining more robust solutions, in terms that they provide more protection against uncertainty, requires willingness to accept higher running times to calculate the optimal solutions.

Tables 2-5 provide more detailed statistics for the four groups of instances and the three algorithmic approaches. On the left hand side, we report statistics on the running times considering only those instances that can be solved to optimality within 500 seconds by all the three approaches. On the right hand side, we provide statistics for the remaining problems (i.e, for those that can not be solved to optimality by at least one of the approaches). For each approach, we report the number of instances



(a) Comparing the three approaches over 16 combinations of $\Gamma_E, \Gamma_V \in \{0, 5, 20, 50\}$.

(b) The Compact approach for four (Γ_E, Γ_V) combinations.

Figure 2: Cumulative percentage of the total number of solved instances of group `C` within 500 seconds.

that are not solved within the time limit (over the total number of instances) and the statistics on the final gap (calculated with respect to the lower bound) over these problems. These statistics on the running times, along with the number of times that an optimal solution cannot be found in less than 500 seconds, indicate that, for the three approaches and across the four family of instances, there is an apparently small number of cases (given by a particular combination of Γ_E and Γ_V) that are intractable under the algorithms used.

Although optimality is not always verified (especially for the Robustness-Cuts approach), the quality of the solutions obtained when reaching the time limit is remarkably good, as it can be seen from the statistics on the final gaps. The values of the median and the mean of the gaps in Tables 2-5 indicate that the chosen formulations and approaches guarantee that solutions of a good quality can be obtained within a reasonable running time, in case that they are not optimal. This observation complements the analysis of Figure 2(b).

	Running times statistics ($t \leq 500$ s)				Gaps (%) statistics ($t > 500$ s)				
Approach	Min	Median	Mean	Max	#	Min	Median	Mean	Max
Compact	0.063	0.969	**6.210**	**275.600**	32/640	0.000	**0.009**	0.038	0.618
R-Cuts	**0.031**	**0.773**	12.380	419.200	63/640	0.000	0.223	0.287	0.817
C-Cuts	0.047	1.320	7.106	318.800	35/640	0.000	0.019	**0.024**	**0.297**

Table 2: Algorithmic performance statistics for group `C`

	Running times statistics ($t \leq 500$ s)				Gaps (%) statistics ($t > 500$ s)				
Approach	Min	Median	Mean	Max	#	Min	Median	Mean	Max
Compact	0.156	5.719	**23.690**	407.200	87/640	0.000	**0.029**	**0.048**	**0.327**
R-Cuts	**0.141**	**4.484**	27.690	476.600	127/640	0.000	0.095	0.132	1.745
C-Cuts	**0.141**	8.297	30.430	**402.600**	96/640	0.000	0.047	0.060	0.609

Table 3: Algorithmic performance statistics for group `D`

Further information about algorithmic performances is presented in the on-line supplement (Tables 8 - 11). The evolution, over time, of the gap between lower and upper bounds in the B&C tree for a subset of the most difficult instances is also analyzed in the supplementary document.

The overall superiority of the Compact approach might be explained by the fact that from the beginning of the optimization process the underlying LP contains complete information regarding the robustness of the solution. Although at the root node we obtain tight bounds even if we consider

	Running times statistics ($t \leq 500$ s)				Gaps (%) statistics ($t > 500$ s)				
Approach	Min	Median	Mean	Max	#	Min	Median	Mean	Max
Compact	0.047	1.156	**10.680**	**133.000**	0/368	**0.000**	**0.000**	**0.000**	**0.000**
R-Cuts	0.047	**0.766**	13.270	476.500	3/368	0.020	0.059	0.064	0.114
C-Cuts	**0.031**	0.719	11.190	197.400	0/368	**0.000**	**0.000**	**0.000**	0.000

Table 4: Algorithmic performance statistics for group K

	Running times statistics ($t \leq 500$ s)				Gaps (%) statistics ($t > 500$ s)				
Approach	Min	Median	Mean	Max	#	Min	Median	Mean	Max
Compact	**0.031**	**0.180**	**0.974**	**8.703**	0/176	0.000	0.000	0.000	0.000
R-Cuts	**0.031**	0.297	4.622	84.200	0/176	0.000	0.000	0.000	0.000
C-Cuts	**0.031**	0.242	1.731	20.380	0/176	0.000	0.000	0.000	0.000

Table 5: Algorithmic performance statistics for group P

Robustness- or Compact-Cuts, after starting the branching process, a large sequence of re-optimizations (each time that a number of cuts is inserted we need to solve the underlying LP) deteriorates the optimization process entailing higher running times. In particular, in the case of Robustness-Cuts the convergence of the values of Θ and Λ becomes slower, i.e., more cuts have to be added and more branch-and-bound nodes have to be enumerated in order to reach optimal values. The combination of these two elements is responsible for the poor performance of this approach with respect to the others. A similar observation is pointed out in [18] when analyzing the performances of the compact formulation and Robustness-Cuts to solve generic MIP problems.

5.3 Results for the Robust B-PCStT

In order to complement the analysis of the computational results presented for the RPCStT, we developed a similar experimental framework for the robust counterpart of the B-PCStT which, as we mentioned before, is an important variant of the PCStT. Because of the restriction on the length of the paper, we only present results obtained for group P and for eleven instances (K100.{0-10}) of group K.

As part of the Robust B-PCStT model, it is necessary to provide a given *budget* B, which represents the maximum allowed sum of the edge costs, considering uncertainty, that the decision maker is willing to pay. Since different instances, even within the same group, have different cost structures, a given value of B might not be suitable for all of them, so it is necessary to establish a fair criterion to define appropriate values of B. In order to do so, we set the budget to be a percentage of a potential *maximum robust budget* value $B_{\max}$, associated with each particular instance. If the input graph is connected, $B_{\max}$ represents the cost of the optimal robust Steiner tree in which all the customers are connected and the cost of at most Γ_E edges is allowed to deviate from its nominal value. If the input graph is not connected, $B_{\max}$ is the cost of the robust Steiner subtree connecting as many customers as possible. To calculate the value of $B_{\max}$, we set the node prizes to a big-M-value and Γ_E to 50, and use one of the algorithms for the RPCStT proposed before. The selected value of $\Gamma_E = 50$ ensures feasibility for all the other values of Γ_E as long as they are not greater than 50, which is the maximum value we consider for this parameter in our experiments. We note that it was necessary to set the node prizes to a big-M-value, instead of simply adding the constraints $y_v = 1 \; \forall v \in V_{p_v^- \geq 0}$ into the MIP model, because the considered instances are not necessarily connected.

B&C Variants. Since there are more alternatives to formulate the Robust B-PCStT as a MIP, there are also more alternatives to design a B&C algorithm. Besides the separation of the connectivity inequalities, we have considered four alternatives to manage the different types of robust constraints: (i) B&C using the compact robust constraints of type (17) and (18) (Compact); (ii) B&C with separation of the robustness cuts of type (27) and type (23) including variable Λ in the objective function (Robustness-Cuts); (iii) B&C with separation of the robust compact constraints of type (17) and (18)

(Compact-Cuts); (iv) B&C with separation of the robustness cuts of type (27) but including all the compact constraints of type (18) (Robustness-Cuts + Compact).

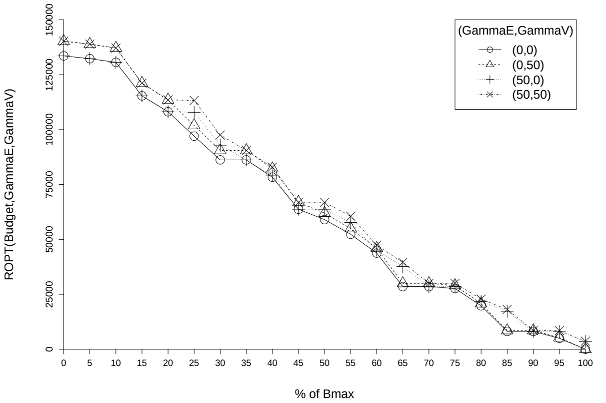
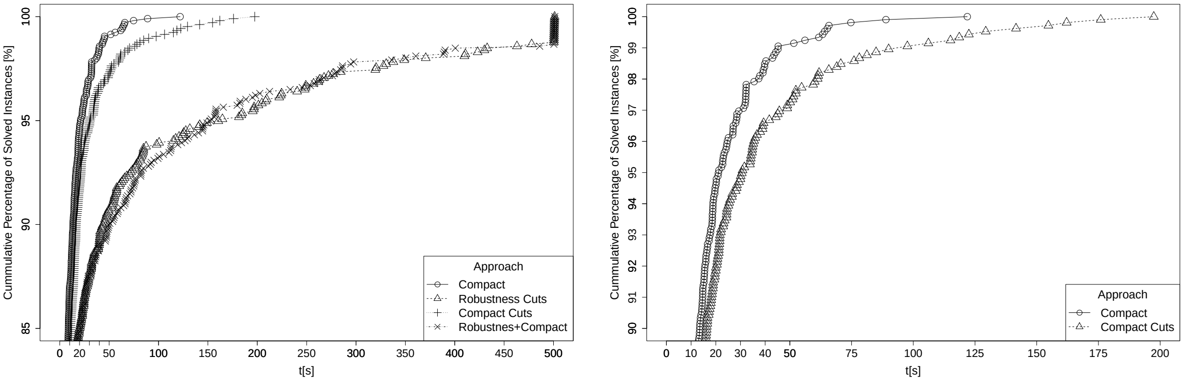


Figure 3: Values of $ROPT_B$ for different values of B, Γ_E and Γ_V, instance `K100.10-0.05-0.05`

The Price of Robustness. In Figure 3 the value of $ROPT_B$ is reported for different values of the budget B and for four different combinations for Γ_E and Γ_V for instance `K100.10-0.05-0.05`. As expected, and independently of the values of Γ_E and Γ_V, there is a monotone decrease of the value of the objective function (recall that this is the sum of the prizes of the nodes that are not connected) when increasing the value of the available budget. When considering a particular value of B, we observe that the differences of $ROPT_B$, among different values of Γ_E and Γ_V, do not present a clear pattern as in the case of the RPCStT. This can be explained by the fact that Γ_E is not included in the objective function but in the budget constraint, so it has an *indirect influence* on the objective function value. For example, when considering a budget given by 25% of $B_{\max}$, the four considered combinations produce significantly different values of $ROPT_B$; while for a budget given by 90% of $B_{\max}$, the four values of $ROPT_B$ are almost the same. Another characteristic that we can observe, is that for tight budgets (0% - 20%) the value of Γ_V has more impact on the model than Γ_E, while for large budgets (80% - 100%) it is just the opposite.

As this was the case for the RPCStT, the latter behaviors are related to the size of the corresponding optimal solution and to its interaction with the problem parameters B, Γ_E and Γ_V. For a tight budget, an optimal solution is made up of only a few edges and many customer nodes are left unconnected, which explains why increasing the value of Γ_V strongly increases the value of the objective function, while increasing Γ_E barely produces changes since only a few edges can be taken into account. On the other hand, for a large budget, most of the customers are connected and an increase of Γ_V might not significantly affect the value of $ROPT_B$, but increasing Γ_E will indeed strongly influence the value of $ROPT_B$ because the budget feasibility will enforce a solution of a smaller cardinality, i.e., it will be necessary to "disconnect" some customers and consequently the value of $ROPT_B$ will be increased. An example that illustrates these dependencies is shown in Figure 10 in the on-line supplement.

Algorithmic Performance. Figure 4(a) shows the cumulative percentage of instances of group `P` solved to optimality within time t with a time limit of 500 seconds, comparing the four approaches described above. In a more detailed plot (see Figure 4(b)), only the Compact and Compact-Cuts approaches are compared. We observe that the Compact and Compact-Cuts approaches are substantially better than the other two approaches (which are both based on the utilization of Robustness-Cuts). For example, to solve 95% percent of the instances of group `P`, the Compact approach needs less than 20 seconds, Compact-Cuts less than 30 seconds, while the other two approaches need almost 150 seconds to solve the same percentage of instances. Moreover, in a small number of cases (less than 2%), the



(a) Comparing all four approaches

(b) Comparing only Compact and Compact-Cuts approaches

Figure 4: Cumulative percentage of the average number of solved instances of group P within $t = 500$ seconds considering different values of $B \in \{0, 5, 10, \dots, 95, 100\} B_{\max}\%$, and $\Gamma_E, \Gamma_V \in \{0, 5, 20, 50\}$.

Robustness-Cuts and Robustness-Cuts + Compact approaches reach the time limit without being able to find optimal solutions within the given time limit.

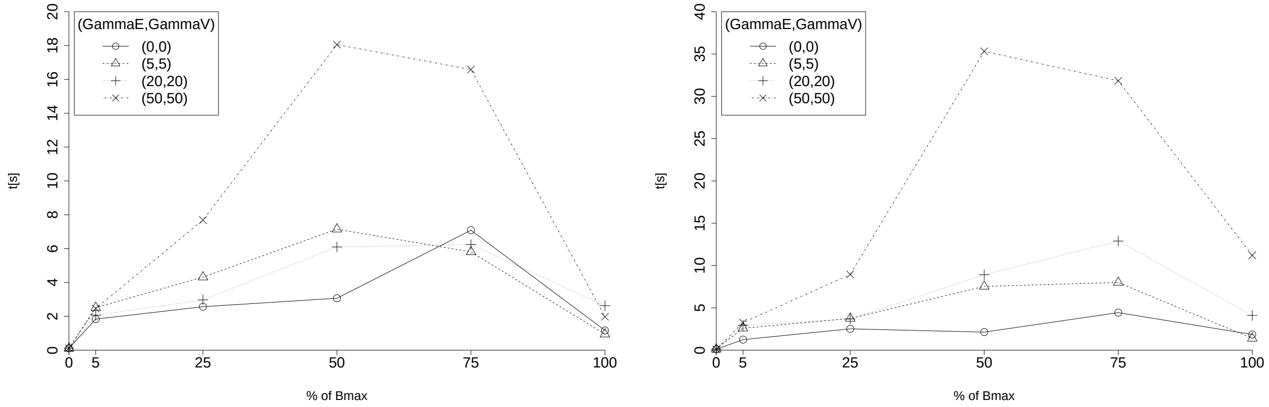
More details about the running times needed to solve the instances, as well as the statistics on the gaps for those instances where at least one of the approaches failed to find an optimal solution, are reported in Table 6. For group P, the Compact approach is the best in terms of average running times. However, the Compact-Cuts approach has a similar performance and provides better minimum and median running times, but a few outliers (see Figure 4(b)) deteriorate the overall statistics. The same table shows that, in 14 out of 1056 cases, the Robustness-Cuts and Robustness-Cuts + Compact approaches do not solve the instances to optimality, but provide very small final gaps.

	Running times statistics ($t \leq 500$ s)				Gaps (%) statistics ($t > 500$ s)				
Approach	Min	Median	Mean	Max	#	Min	Median	Mean	Max
Compact	0.031	0.500	**3.621**	**65.831**	**0**/1056	**0.000**	**0.000**	**0.000**	**0.000**
R-Cuts	0.031	0.546	16.450	477.900	14/1056	0.000	0.417	0.671	1.811
C-Cuts	**0.015**	**0.453**	4.791	162.100	**0**/1056	**0.000**	**0.000**	**0.000**	**0.000**
R-Cuts+C	**0.015**	0.546	16.630	486.000	14/1056	0.000	0.811	0.836	2.332

Table 6: Algorithmic performance statistics for group P (Robust B-PCStT).

Comparing the statistics for the RPCStT (see Table 5) with the results presented in Table 6 for the Robust B-PCStT, we may conclude that the Robust B-PCStT is a considerably more complex problem. With an inclusion of a budget constraint the search for an optimal solution becomes a more difficult numeric task. The influence of the budget level on the algorithmic performance is shown in Figure 5, where the average running times over all the instances of group P are displayed for different budget levels and different values of Γ_E and Γ_V. Our first observation is that, independently of the values of the budget, increasing values of Γ_E and Γ_V directly influence the running times as it was the case of the RPCStT (see Figure 2). However, budgets levels set between $[0\%, 25\%]$ or $[75\%, 100\%]$ entail a better algorithmic performance than those taken from $[25\%, 75\%]$, and the influence of Γ_E and Γ_V is more accentuated in the latter case.

These relations between the running times and the budget levels can be explained by the way how different values of B reduce the space of feasible solutions. Tight budgets, let us say $[0\%, 25\%]$, strongly limit the set of feasible solutions, i.e., they usually correspond to small trees connecting a few customer nodes. Therefore, and considering that most of the P instances are sparse graphs (see Table 7 in the online supplement), the optimal solutions can be quickly obtained. On the other hand, optimal solutions for large budgets, as those defined by $[75\%, 100\%]$, will be usually comprised by almost all the customer nodes; hence, solutions will be similar to the robust Steiner tree connecting those customers, which explains the decrease of the running times. On the contrary, for B chosen from $[25\%, 75\%]$ of $B_{\max}$, the combinatorial nature of the problem seems to have more influence on the algorithmic performance and there are more solutions, probably each of them with a very different structure, that might verify the optimality. Consequently, the computational effort to find an optimal solution is greater as illustrated in Figure 5.



(a) Compact approach

(b) Compact-Cuts approach

Figure 5: Average running times for group P for different values of $B \in \{0, 5, 10, \ldots, 95, 100\}B_{\max}$, and the four selected combinations of (Γ_E, Γ_V).

A similar analysis of the results obtained for instances K100.{0-10} (see Figure 9 and Table 12 in the on-line supplement) lead us to conclude that the Compact-Cuts approach is the best one, both in terms of average and median running times. Once more, the Robustness-cuts based approaches do not seem to be competitive although they solve to optimality all instances within the given time limit. Consequently, we may say that both the Compact and Compact-Cuts approaches are the most effective ones for solving the robust B-PCStT for the considered instances.

6. Improved B&S Algorithms for the RPCStT and its Variants

Although in this paper we presented MIP-based exact approaches for solving the robust counterparts of the PCStT and of its variants, it is also possible to solve them by successively solving a finite number of classical instances of the corresponding problem. The next corollaries are derived from the more general results presented in [3]. To apply the results below, we assume that the customers and the edges are sorted in non-increasing order with respect to their deviations, i.e., $d_1 \geq d_2 \geq d_3 \ldots$ and the last deviations $d_{n'+1}$ (for the customers) and d_m (for the edges) are set to zero.

Lemma 1. *Given $\Gamma_E \in \{0, \ldots, m\}$ and a given $\Gamma_V \in \{0, \ldots, n'\}$, the B&S Robust Counterpart of the GW-PCStT can be solved by solving $(n' - \Gamma_V + 2)(m - \Gamma_E + 2)$ nominal problems*

$$ROPT(\Gamma_E, \Gamma_V) = \min_{\substack{a \in \{\Gamma_E, \ldots, m+1\} \\ b \in \{\Gamma_V, \ldots, n'+1\}}} G^{a,b},$$

where for $a \in \{\Gamma_E, \ldots, m+1\}$ and $b \in \{\Gamma_V, ..., n'+1\}$:

$$G^{a,b} = \Gamma_E d_a + \Gamma_V d_b + \min_{(\mathbf{x},\mathbf{y}) \in \mathscr{T}} \left(\sum_{e \in E} c_e^- x_e + \sum_{e=1}^{a} (d_e - d_a) x_e + \sum_{v \in V} p_v^- (1 - y_v) + \sum_{v=1}^{b} (d_v - d_b)(1 - y_v) \right).$$

Lemma 2. *Given $\Gamma_E \in \{0,\ldots,m\}$ and $\Gamma_V \in \{0,\ldots,n'\}$, the Robust B&S B-PCStT can be solved by solving $(n'-\Gamma_V+2)(m-\Gamma_E+2)$ nominal problems*

$$ROPT_B\left(\Gamma_E,\Gamma_V\right) = \min_{\substack{a\in\{\Gamma_E,\ldots,m+1\}\\ b\in\{\Gamma_V,\ldots,n'+1\}}} G_B^{a,b}, \tag{28}$$

where for $a \in \{\Gamma_E,...,m+1\}$ and $b \in \{\Gamma_V,...,n'+1\}$

$$G^{a,b} = \Gamma_V d_b + \min_{(\mathbf{x},\mathbf{y})\in\mathscr{T}} \left(\sum_{v\in V} p_v^-(1-y_v) + \sum_{v=1}^{b}(d_v-d_b)(1-y_v) \,\middle|\, \sum_{e\in E} c_e^- x_e + \sum_{e=1}^{a}(d_e-d_a)x_e + \Gamma_E d_a \le B. \right)$$

Lemma 3. *Given $\Gamma_E \in \{0,\ldots,m\}$ and $\Gamma_V \in \{0,\ldots,n'\}$, the Robust B&S Q-PCStT can be solved by solving $(m-\Gamma_E+2)(n'-\Gamma_V+2)$ nominal problems*

$$ROPT_Q\left(\Gamma_E,\Gamma_V\right) = \min_{\substack{a\in\{\Gamma_E,\ldots,m+1\}\\ b\in\{\Gamma_V,\ldots,n'+1\}}} G_Q^{a,b}, \tag{29}$$

where for $a \in \{\Gamma_E,...,m+1\}$ and $b \in \{\Gamma_V,...,n'+1\}$

$$G^{a,b} = \Gamma_E d_a + \min_{(\mathbf{x},\mathbf{y})\in\mathscr{T}} \left(\sum_{e\in E} c_e^- x_e + \sum_{e=1}^{a}(d_e-d_a)x_e \,\middle|\, \sum_{v\in V} p_v^-(1-y_v) + \sum_{v=1}^{b}(d_v-d_b)(1-y_v) + \Gamma_V d_b \le Q. \right)$$

Lemmas 1-3 are particulary useful when polynomial-time algorithms are available for graphs with some special structures (see, e.g., Corollary 1). We want to point out that for the case of general graphs, where the deterministic counterparts are NP-Hard, branch-and-cut algorithms like the ones presented in this paper remain a preferable option.

Corollary 1. *If the input graph is a tree, a series-parallel graph or a 2-tree, the robust counterpart of the PCStT can be solved in $O(|V|^3)$ time.*

Proof. The deterministic PCStT can be solved in $O(|V|)$ time on trees (see [27]). A series-parallel graph can be completed in linear time into a 2-tree. [1] have shown that the PCStT can be solved in $O(|V|)$ time on 2-trees. We complete the proof by combining these results with the result of Lemma 1. □

7. Conclusions and Future Work

In this paper we studied the PCStT and its budget and quota constrained variants assuming interval uncertainty associated with their input parameters. To include and handle this uncertainty we considered the B&S robust optimization (RO) approach, formulating the robust counterpart of the problems by means of different mixed integer problems. Specific branch-and-cut algorithms were implemented to solve these problems. The algorithms were tested on a set of benchmark instances generated from state-of-the-art instances of the deterministic version of the problem. The obtained computational results suggest that: (1) the RO model allows to produce different robust solutions for different levels of conservatism. These solutions provide a protection in terms of the relatively small increase of the solution cost in presence of an increased uncertainty. This important feature of the model offers to the decision maker more flexibility to choose a solution according to her/his perception of the uncertain state of the decision-making environment. (2) The algorithmic performance strongly depends on the model parameters, Γ_E and Γ_V (and B in the case of the Robust B-PCStT). There is a strong correlation between the size of the optimal solution and the corresponding values for B, Γ_E and Γ_V. (3) Among three possibilities to deal with robustness in a MIP model, the addition of a compact set of constraints right at the beginning of the Branch-and-Bound process, outperforms the remaining two (cutting planes) approaches. This can be explained by the fact that from the beginning of the optimization process the underlying LP contains complete information regarding the robustness of the solution, which allows CPLEX to exploit its powerful preprocessing, heuristics and MIP algorithms, while this is not possible for the cutting plane approaches.

As possible directions for future work, it would be interesting to develop algorithms for 2-trees (or, graphs with a bounded tree-width, in general) that improve the trivial running times obtained by running $O(|V||E|)$ iterations of the deterministic problem. In addition, a strategy combining the results described in Section 6 and the utilization of further polyhedral techniques might improve the results we obtained in terms of algorithmic performance.

Acknowledgements. I.Ljubić is supported by the APART Program of the Austrian Academy of Sciences. This support is greatly acknowledged. E. Álvarez-Miranda thanks the Institute of Advances Studies of the Università di Bologna from where he is PhD Fellow.

References

[1] E. Álvarez-Miranda, A. Candia, X. Chen, B. Li, and X. Hu. Efficient algorithms for the PCStT with interval data. *AAIM 2010*, LNCS 6124:13–24, 2010.

[2] E. Álvarez-Miranda, I. Ljubić, and P. Toth. Exact solutions for the robust prize-collecting Steiner tree problem. *Proceedings of the 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, October 5-7 2011, Budapest, Hungary*, ICUMT 2011:1–7, 2011.

[3] E. Álvarez-Miranda, I. Ljubić, and P. Toth. A note on the Bertsimas and Sim algorithm for robust combinatorial optimization problems. Technical report, Department of Statistics and Operations Research, University of Vienna, 2012. Submitted.

[4] A. Archer, M. Bateni, M. Hajiaghayi, and H. Karloff. Improved approximation algorithms for prize-collecting Steiner tree and TSP. *SIAM Journal on Computing*, 40:309–332, 2011.

[5] M. Bailly-Bechet, A. Braunstein, and R. Zecchina. A prize-collecting Steiner tree approach for transduction network inference. *CMSB 2009*, LNBI 5688:83–95, 2009.

[6] M. Bailly-Bechet, C. Borgs, A. Braunstein, J. Chayes, A. Dagkessamanskaia, J.-M. Franois, and R. Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *PNAS*, 108:882–887, 2011.

[7] E. Balas. The prize-collecting traveling salesman problem. *Networks*, 19:621–636, 1989.

[8] J.E. Beasly. Collection of test data sets for a variety of operations research (OR) problems, 1990. URL `http://people.brunel.ac.uk/ mastjjb/jeb/info.html`.

[9] A. Ben-Tal, L. El-Ghaoui, and A. Nemirovski, editors. *Robust Optimization.* Princeton Series in Applied Mathematics, 1st edition, 2010.

[10] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Math. Prog.*, Series B 98:49–71, 2003.

[11] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

[12] D. Bienstock, M. X. Goemans, D Simchi-Levi, and D. Williamson. A note on the prize collecting traveling salesman problem. *Mathematical Progamming*, 59:413–420, 1993.

[13] S. A. Canuto, M. G. C. Resende, and C. C. Ribeiro. Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, 38:50–58, 2001.

[14] X. Chen, J. Hu, and X. Hu. A new model for path planning with interval data. *Computers & OR*, 36:1893–1899, 2009.

[15] B.V. Cherkassky and A.V. Goldberg. On implementing a push-relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1997.

[16] M. Dittrich, G. Klau, A. Rosenwald, T. Dandekar, and T. Muller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *ISMB 2008*, 24:223–231, 2008.

[17] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22:425–460, 2000.

[18] M. Fischetti and M. Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation*, 4(3):239–273, 2012.

[19] M.X. Goemans and D.P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In D. S. Hochbaum, editor, *Approximation algorithms for NP-hard problems*, chapter 4, pages 144–191. PWS Publishing Co., Boston, MA, USA, 1997.

[20] M Haouari and J. C. Siala. A hybrid lagrangian genetic algorithm for the prize collecting Steiner tree problem. *Computers & OR*, 33:1274–1288, 2006.

[21] M. Haouari, S. Layeb, and H. Sherali. The prize collecting steiner tree problem: models and lagrangian dual optimization approaches. *Computational Optimization and Applications*, 40:13–39, 2008.

[22] M. Haouari, S. Layeb, and H. Sherali. Algorithmic expedients for the prize collecting Steiner tree problem. *Discrete Optimization*, 7:32–47, 2010.

[23] S. Huang. *A constraint optimization framework for discovery of cellular signaling and regulatory networks*. PhD thesis, Massachusetts Institute of Technology, June 2011.

[24] S. Huang and E. Fraenkel. Integration of proteomic, transcriptional, and interactome data reveals hidden signaling components. *Science Signaling*, 2:ra40, 2009.

[25] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *BIOINFORMATICS*, 18:233–240, 2002.

[26] D. S. Johnson, M. Minkoff, and S. Phillips. The prize-collecting Steiner tree problem: Theory and practice. *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms, 9-11 January 2000, San Francisco, USA*, SODA 2000:760–769, 2000.

[27] G. Klau, I. Ljubić, P. Mutzel, U. Pferschy, and R. Weiskircher. The fractional prize-collecting Steiner tree problem on trees. *ESA 2003*, LNCS 2832:691–702, 2003.

[28] G. Klau, I. Ljubić, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, and R. Weiskircher. Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problems. *GECCO 2004*, LNCS 3102:1304–1315, 2004.

[29] T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32: 207–232, 1998.

[30] C. Lasher, C. Poirel, and T. Murali. Cellular response networks. In L. Heath and N. Ramakrishnan, editors, *The Problem Solving Handbook for Computational Biology and Bioinformatics*, pages 233–250. Springer-Verlag, 2010.

[31] I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Math. Prog.*, Series B 105:427–449, 2006.

[32] A. Lucena and M. G. Resende. Strong lower bounds for the prize-collecting Steiner problem in graphs. *Discrete Applied Mathematics*, 141:277–294, 2004.

[33] R. Montemmani, J. Barta, M. Mastrolilli, and L. Gambardella. The robust traveling salesman problem with interval data. *Transportation Science*, 41:366–381, 2007.

[34] A. Prodon, S. DeNegre, and T. M. Liebling. Locating leak detecting sensors in a water distribution network by solving prize-collecting Steiner arborescence problems. *Mathematical Programming*, Series B 124:119–141, 2010.

[35] A. Salles da Cunha, A. Lucena, N. Maculan, and M. G. C. Resende. A relax-and-cut algorithm for the prize-collecting Steiner problem in graphs. *Discrete Applied Mathematics*, 157:1198–1217, 2009.

[36] A. Segev. The node-weighted Steiner tree problem. *Networks*, 17:1–17, 1987.

[37] E. Uchoa. Reduction tests for the prize-collecting Steiner problem. *Operations Research Letters*, 34:437–444, 2007.

[38] S.P. Uryasev and P.M. Pardalos, editors. *Stochastic Optimization: Algorithms and Applications.* Springer (Applied Optimization Volume 54), 1st edition, 2001.

[39] S. Vijayanarasimhan and K. Grauman. Efficient region search for object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:1401–1408, 2011.